

#2

PATENTS

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

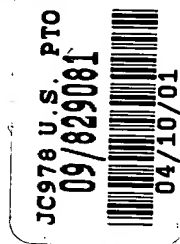
In re application of

Kazumasa TODA

Serial No. (unknown)

Filed herewith

PROCESSOR SYSTEM



CLAIM FOR FOREIGN PRIORITY UNDER 35 U.S.C. 119
AND SUBMISSION OF PRIORITY DOCUMENT

Assistant Commissioner for Patents

Washington, D.C. 20231

Sir:

Attached hereto is a certified copy of applicant's corresponding patent application filed in Japan on April 11, 2000 under No. 2000-109415.

Applicant herewith claims the benefit of the priority filing date of the above-identified application for the above-entitled U.S. application under the provisions of 35 U.S.C. 119.

Respectfully submitted,

YOUNG & THOMPSON

By

Robert J. Patch
Attorney for Applicant
Registration No. 17,355
745 South 23rd Street
Arlington, VA 22202
Telephone: 703/521-2297

April 10, 2001

日 本 国 特 許 庁
PATENT OFFICE
JAPANESE GOVERNMENT

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日

Date of Application:

2000年 4月11日

出 願 番 号

Application Number:

特願2000-109415

出 願 人

Applicant (s):

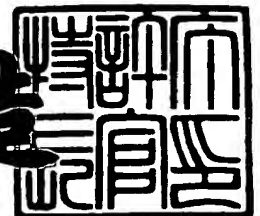
北陸日本電気ソフトウェア株式会社



2001年 2月 9日

特許庁長官
Commissioner,
Patent Office

及川耕造



出証番号 出証特2001-3006175

【書類名】 特許願

【整理番号】 10601711

【提出日】 平成12年 4月11日

【あて先】 特許庁長官 殿

【国際特許分類】 G06F 9/46
G06F 15/16

【発明者】

【住所又は居所】 石川県石川郡鶴来町安養寺1番地 北陸日本電気ソフトウェア株式会社内

【氏名】 戸田 和成

【特許出願人】

【識別番号】 000242666

【氏名又は名称】 北陸日本電気ソフトウェア株式会社

【代理人】

【識別番号】 100088328

【弁理士】

【氏名又は名称】 金田 暢之

【電話番号】 03-3585-1882

【選任した代理人】

【識別番号】 100106297

【弁理士】

【氏名又は名称】 伊藤 克博

【選任した代理人】

【識別番号】 100106138

【弁理士】

【氏名又は名称】 石橋 政幸

【手数料の表示】

【予納台帳番号】 089681

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9714328

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 プロセッサシステム

【特許請求の範囲】

【請求項 1】 第一のアーキテクチャによる第一のオペレーティングシステムが動作するプロセッサシステム上に、第二のアーキテクチャによる第二のオペレーティングシステムを前記第一のオペレーティングシステムと共に動作させるヘテロジーニアスなプロセッサシステムであって、

前記プロセッサシステムのハードウェア資源を、前記第一のオペレーティングシステムにて使用するハードウェア資源と、前記第二のオペレーティングシステムにて使用するハードウェア資源とに分割する手段と、

分割する前記ハードウェア資源の種類及び数量を記憶する手段と、

前記第二のオペレーティングシステム用に確保した前記ハードウェア資源にて前記第二のオペレーティングシステムを起動する手段と、

前記第一のオペレーティングシステム用に確保した前記ハードウェア資源にて、前記第一のオペレーティングシステムを起動する手段とを有し、

前記第一のオペレーティングシステムと前記第二のオペレーティングシステムとを同時に独立に動作可能としたプロセッサシステム。

【請求項 2】 前記ハードウェア資源を分割する前に分割する前記ハードウェア資源の種類及び数量を変更する手段を有し、システム起動時に使用するハードウェア資源を変更できる請求項 1 に記載のプロセッサシステム。

【請求項 3】 前記第一のオペレーティングシステムと前記第二のオペレーティングシステムとで共有できるメモリ領域を確保する手段を有し、メモリを介して前記第一のオペレーティングシステムと前記第二のオペレーティングシステムとのデータ交換が行える請求項 1 または請求項 2 に記載のプロセッサシステム。

【請求項 4】 前記第一のオペレーティングシステムと前記第二のオペレーティングシステムとで通信を行う手段を有し、共通メモリ領域を使用して前記第一のオペレーティングシステムと前記第二のオペレーティングシステムとが通信を行うことにより、前記第一のオペレーティングシステムが動作するシステムを

前記第二のオペレーティングシステムが動作するシステムの入出力プロセッサとして利用する請求項 3 に記載のプロセッサシステム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明はあるアーキテクチャによるオペレーティングシステムが動作するプロセッサシステム上に、そのオペレーティングシステムと同時に独立して別のアーキテクチャによるオペレーティングシステムを動作させるプロセッサシステムに関する。

【0002】

【従来の技術】

近年、特定のメーカーに依存しない廉価なハードウェアとその上で動作するオペレーティングシステムの機能および性能の向上に伴い、高価な汎用コンピュータによる従来の中央集中型のシステムをクライアント・サーバ型のオープンシステムによって置き換える傾向が高くなってきている。

【0003】

しかしながら、従来の汎用コンピュータによるプロプライエタリなシステムを一度にオープンなシステムに置き換える作業は多大な期間とコストを要し、容易に移行できない場合が多い。

【0004】

そのため、一旦オープンシステムと汎用コンピュータとを融合させ、両者の長所を取り入れたシステムを構築し、段階的にオープンなシステムに移行していく手法が取られる。

【0005】

オープンシステムと汎用コンピュータとを融合させる方法としては、第一にオープンシステムと汎用コンピュータをローカルエリアネットワークなどの外部ネットワークで接続してデータを交換する方法があり、

第二に、オープンシステムのノードと汎用コンピュータのノードを一つのハードウェアの中に組み込み、特殊なハードウェアによる内部的なネットワークで接

続してデータを交換する方法があり、

第三に、オープンシステムのサーバ上のミドルウェアとして汎用コンピュータのオペレーティングシステムをエミュレーションする方法などがある。

【 0 0 0 6 】

第一の方法は、従来の技術を組み合わせるだけで実現できるが、外部ネットワーク接続の通信にオーバーヘッドが伴うためデータ交換の高速化には限度がある。また、高価な汎用コンピュータを引き続き使用することになる。

【 0 0 0 7 】

第二の方法は、内部的なネットワークを使用することによりデータ交換を高速化することはできるが、内部的な接続のために特殊なハードウェアを必要とするとともに、高価な汎用コンピュータの中心的なハードウェアも必要とする。

【 0 0 0 8 】

第三の方法は、オープンシステムだけで特殊なハードウェアや外部ネットワークなどを使用することなく廉価に実現できるが、汎用コンピュータのオペレーティングシステムをオープンなオペレーティングシステム上でミドルウェアとして動作させるためには、オープンなオペレーティングシステムの振る舞いに合わせて汎用コンピュータのオペレーティングシステムを大幅に修正する必要がでてくる欠点がある。

【 0 0 0 9 】

【発明が解決しようとする課題】

廉価なシステムへの移行段階としてオープンシステムと汎用コンピュータを融合したシステムを実現する上で、高価な汎用コンピュータを引き続き使用することによりシステム価格を低下させにくくしていることは問題である。

【 0 0 1 0 】

また、廉価にオープンなシステムと汎用コンピュータを融合したシステムを実現するために、オープンシステムのオペレーティングシステム上でエミュレーションすることで従来のシステムを大幅に変更しなければならなくなっている問題である。

【 0 0 1 1 】

本発明は前記課題を解決するものであり、特定のメーカーに依存しない廉価なハードウェアとソフトウェアを使用したオープンシステムのみを使用して、オープンシステムと高速に動作する汎用コンピュータ環境を融合したシステムを実現するヘテロジーニアスなプロセッサシステムを提供することを目的とする。

【 0 0 1 2 】

【課題を解決するための手段】

上記目的を達成するために、請求項 1 の発明にかかるプロセッサシステムは、第一のオペレーティングシステムを起動する前に第一のオペレーティングシステムで使用するハードウェア資源と第二のオペレーティングシステムで使用するハードウェア資源とを分割する手段と、分割するハードウェアの種類及び数量を記憶しておく手段と、第二のオペレーティングシステム用に確保したハードウェア資源にて第二のオペレーティングシステムを起動する手段と、第一のオペレーティングシステム用に確保したハードウェア資源にて第一のオペレーティングシステムを起動する手段とを有し、第一のオペレーティングシステムが起動する前に第二のオペレーティングシステムに必要なハードウェア資源を確保して、第一のオペレーティングシステムと第二のオペレーティングシステムとを同時に独立して動作させるようにしたものである。

【 0 0 1 3 】

また、請求項 2 の発明にかかるプロセッサシステムは、第一のオペレーティングシステムで使用するハードウェア資源と第二のオペレーティングシステムで使用するハードウェア資源の種類及び数量をシステム起動毎に変更できるようにしたものである。

【 0 0 1 4 】

また、請求項 3 の発明にかかるプロセッサシステムは、メモリを介して第一のオペレーティングシステムと第二のオペレーティングシステムが通信できるようにしたものである。

【 0 0 1 5 】

また、請求項 4 の発明にかかるプロセッサシステムは、外部入出力機器との入出力処理を第二のオペレーティングシステムで行わずに第一のオペレーティング

システムで行うことにより、第二のオペレーティングシステムの開発量を削減するとともに第二のオペレーティングシステムから容易に第一のオペレーティングシステムで使用する最新の外部入出力機器を利用できるようにしたものである。

【0016】

【発明の実施の形態】

(本発明の第一の実施の形態)

以下に、本発明の第一の実施の形態を図を用いて詳細に説明する。図1においては、10はオープンシステムのコンピュータで、プロセッサ20、メモリ30およびシステム起動時に最初に動作するファームウェアを格納している記憶装置40、電源を切っても内容が保持される不揮発性メモリ45を備えている。また、オープンシステムのコンピュータ10にはキーボード50、ディスプレイ60、ディスク70、外部入出力装置80が接続されている。

【0017】

また、図2において分割ハードウェア資源数量記憶手段1によって得られる分割ハードウェア資源管理テーブル100の情報によりハードウェア資源分割手段2はハードウェア資源管理テーブル200から第二のオペレーティングシステムで使用するハードウェア資源のエントリを削除し、そのエントリを第二オペレーティングシステム用ハードウェア資源管理テーブル300に格納する。これにより、第二のオペレーティングシステムで使用するハードウェア資源が第一のオペレーティングシステムの管理から外れることになり、ハードウェア資源を第一のオペレーティングシステム用と第二のオペレーティングシステム用に論理的に分割する。

【0018】

本発明の例によると図3に示すように分割ハードウェア資源管理テーブル100は第一のオペレーティングシステム管理下から第二のオペレーティングシステム管理下へ移行させるハードウェア資源の種類と数量を格納している。システムの電源が切れても情報を保持するために分割ハードウェア資源管理テーブル100は不揮発性メモリ45またはディスク70の特定領域に格納される。

【0019】

図4に示すハードウェア資源管理テーブル200はハードウェア資源毎のテーブルを持ち、システム起動時に接続されているハードウェア資源に基づいてメモリ30の中のファームウェアが管理する領域にファームウェアによって作成される。本発明の例によるとハードウェア資源管理テーブル200はプロセッサ、割り込み、メモリに関するテーブルを持っている。図5によるとプロセッサを管理するプロセッサ管理テーブル210はプロセッサIDとマスタであるかスレーブであるかを示す属性を含んでいる。図6によると割り込みを管理するテーブル割り込み管理テーブル220は装置を特定するデバイスIDと割り込み元の割り込み番号を示す割込番号とその割り込みを処理するプロセッサを示す通知先とレベルトリガかエッジトリガかなど割り込みの契機を示す属性を含んでいる。図7によるとメモリを管理するメモリ管理テーブル230は開始アドレスとレンジと使用可能かシステムによって予約されているかを示す属性を含んでいる。

【0020】

本発明の例ではハードウェア資源分割手段2により、プロセッサ20は第一のオペレーティングシステムであるオープンシステムのオペレーティングシステム用のプロセッサ21と第二のオペレーティングシステムである汎用コンピュータをエミュレートするエミュレートソフトウェア用のプロセッサ22に論理的に分割され、メモリ30は第一のオペレーティングシステム用のメモリ31と第二のオペレーティングシステム用のメモリ32に論理的に分割される。また、接続されている外部装置もキーボード50、ディスプレイ60、ディスク70は第一のオペレーティングシステム用に、外部入出力装置80は第二のオペレーティングシステム用に論理的に分割される。最終的にオープンシステムのコンピュータ10は第一オペレーティングシステム部11と第二オペレーティングシステム部12に論理的に分割される。

【0021】

図8に示す第二オペレーティングシステム用ハードウェア資源管理テーブル300はハードウェア資源管理テーブル200と同様の構造をしており、第二オペレーティングシステムで使用するハードウェア資源に関する情報を格納している。ハードウェア資源分割手段2によりメモリ32の特定の領域に作成されるもの

で、本発明の例ではメモリ 3 2 の先頭領域に作成されるものとする。

【 0 0 2 2 】

第二オペレーティングシステム用ハードウェア資源管理テーブル 3 0 0 の情報に基づいて第二オペレーティングシステム起動手段 3 はメモリ 3 2 の第二オペレーティングシステム用ハードウェア資源管理テーブル 3 0 0 以外の領域に第二オペレーティングシステムをロードし、第二オペレーティングシステム用プロセッサで起動させる。第一オペレーティングシステム起動手段 4 は第一オペレーティングシステムのローダを起動することにより第一オペレーティングシステムを起動する。

【 0 0 2 3 】

次に、図 9 のフローチャートを参照しながら従来のオープンシステムの起動時の動作を説明する。

【 0 0 2 4 】

オープンシステムの電源投入時、複数のプロセッサが接続されている場合、特定のアルゴリズムによりマスタとなるプロセッサが決定され（ステップ A 1）、その他のプロセッサはスレーブとしてマスタからの起動通知を待つ待機状態となる（ステップ A 2）。マスタプロセッサでは、記憶装置 4 0 に格納されているファームウェア処理を開始する。このときオペレーティングシステムに対して接続されているハードウェア資源の情報を引き渡すためのテーブルであるハードウェア資源管理テーブル 2 0 0 をメモリ 3 0 の特定のアドレスに作成する（ステップ A 3）。また、そのハードウェアが正常かどうか確認するための試験や初期化処理が行われる（ステップ A 4）。その後、オペレーティングシステムを起動するためにディスク 7 0 の特定の領域にある初期プログラムローダをメモリ 3 0 の特定のアドレスに展開して実行する（ステップ A 5）。この初期プログラムローダはディスク 7 0 上のオペレーティングシステムのローダをメモリ 3 0 のハードウェア資源管理テーブル 2 0 0 とは別のアドレスに展開して、ハードウェア資源管理テーブル 2 0 0 の情報を元に仮想メモリを実現するためのページテーブルの作成や割り込みを取り扱うための割り込み管理テーブルなどを作成した後、待機状態のスレーブプロセッサを起動して（ステップ A 6）、オペレーティングシステ

ムを稼働する（ステップA7）。

【0025】

これに対して本発明の一例では初期プログラムローダが第一のオペレーティングシステムのローダをメモリに展開して実行する前に、第一のオペレーティングシステムから分割するハードウェア資源の数や容量を記憶しておく手段から得られる情報に基づいて第一のオペレーティングシステムからハードウェア資源を分割する手段によってプロセッサ22とメモリ32と外部入出力装置80をファームウェアが作成したハードウェア資源を管理するテーブルから削除する。次にディスク70に格納されている第二のオペレーティングシステムのイメージをメモリ32に展開し、プロセッサ22で実行させる。最後に第一のオペレーティングシステムであるオープンシステムのオペレーティングシステムのローダをメモリに展開して実行することによりヘテロジーニアスな環境を構築する。

【0026】

具体的には図10のフローチャートを参照しながら説明する。

【0027】

電源投入後、従来のオープンシステムと同様にマスタとなるプロセッサ、スレーブとなるプロセッサを決定する（ステップB1）。このときスレーブプロセッサはマスタからの起動通知を待つ待機状態となる（ステップB2、B3）。従来と同様にマスタプロセッサでは、記憶装置40に格納されているファームウェア処理を開始し、ハードウェア資源管理テーブル200の作成（ステップB4）、ハードウェアの試験や初期化（ステップB5）を行う。次に分割ハードウェア資源数量記憶手段1が分割ハードウェア資源管理テーブル100から分割するハードウェア資源の数量を取得する（ステップB6）。分割ハードウェア資源管理テーブル100は図3に示す構造を持ち、第二のオペレーティングシステム用に確保するハードウェア資源の数量を記憶しておくテーブルで、ディスク70の特定のエリアまたは不揮発性メモリ45に格納される。この情報に基づいてハードウェア資源分割手段2がハードウェア資源管理テーブル200から第二のオペレーティングシステム用のハードウェア資源のエントリを削除または変更し、ハードウェア資源を第一のオペレーティングシステムから分割する（ステップB7）。

【 0 0 2 8 】

具体的には図 3 に示すように第二オペレーティングシステム用プロセッサエントリ 1 0 1 に「2」が格納されているので、プロセッサ管理テーブル 2 1 0 のエントリからプロセッサ ID の大きい 2 つのエントリ 2 1 3 と 2 1 4 を削除する（図 5 の 1 → 2）。また、図 3 に示すように第二オペレーティングシステム用割り込みエントリ 1 0 2 に「8 0」が格納されているので、割り込み管理テーブル 2 2 0 のエントリからデバイス ID 8 0 のエントリ 2 2 3 を削除する（図 6 の 1 → 2）。また、図 3 に示すように第二オペレーティングシステム用メモリエントリ 1 0 3 に「2 5 6 MB」が格納されているので、メモリ管理テーブル 2 3 0 のエントリのうち 2 5 6 MB 以上のメモリを管理しているエントリ 2 3 3 のレンジ情報を書き換える（図 7 の 1 → 2）。これによりプロセッサを 2 つ、割り込みを 1 つ、メモリ 2 5 6 MB を第一のオペレーティングシステムが管理できないハードウェア資源として第二のオペレーティングシステム用に分割することになる。このときマスタプロセッサはプロセッサ 2 1 に含まれ、プロセッサ 2 2 はスレーブプロセッサとして待機していることとなる。また、ハードウェア資源分割手段 2 は第二のオペレーティングシステムで使用するハードウェア資源を管理する第二オペレーティングシステムハードウェア資源管理テーブル 3 0 0 をメモリ 3 2 の先頭領域に作成する。

【 0 0 2 9 】

次にディスク 7 0 の特定の領域から第二のオペレーティングシステムをメモリ 3 2 の第二オペレーティングシステムハードウェア資源管理テーブル 3 0 0 以外の領域に読み込み（ステップ B 8）、プロセッサ 2 2 に対して第二オペレーティングシステムの開始アドレスをパラメータとした起動通知を発行し、第二オペレーティングシステム用プロセッサで第二のオペレーティングシステムを起動する（ステップ B 9）。第二のオペレーティングシステムは第二オペレーティングシステムハードウェア資源管理テーブル 3 0 0 の情報に基づいて仮想メモリを実現するためのページテーブルや割り込みを取り扱うための割り込み管理テーブルなどを作成して動作を開始する（ステップ B 1 0）。同時に、第一オペレーティングシステム起動手段 4 により第一のオペレーティングシステムを起動するために

ディスク 70 の特定の領域にある初期プログラムローダをメモリ 30 の特定のアドレスに展開して実行する（ステップ B 1 1）。この初期プログラムローダは従来技術そのままのものであり、ディスク 70 上の第一のオペレーティングシステムのローダをメモリ 30 に展開して、ファームウェアが作成したハードウェア資源管理テーブルの情報を元に仮想メモリを実現するためのページテーブルの作成や割り込みを取り扱うための割り込み管理テーブルなどを作成する。この時点でステップ B 7 で第二のオペレーティングシステム用に確保したハードウェア資源は第一のオペレーティングシステムからは認識できないものとなっている。その後、待機状態のスレーブプロセッサを起動して（ステップ B 1 2）、第一のオペレーティングシステムを稼働する（ステップ B 1 3）。

【 0 0 3 0 】

ここで、ステップ B 6 から B 1 1 までの処理はファームウェア処理として記憶装置 40 に格納しておくことも可能であるが、実行可能なプログラムイメージとしてディスク 70 に格納しておき、ステップ B 6 の処理でこの実行可能なプログラムをロードして実行する方が望ましい。なぜなら、記憶装置 40 に格納されているファームウェアはハードウェアに依存した処理が多く含まれているため、ハードウェアの一部としてデータ処理装置毎に作成されることになる。これに対し実行可能なプログラムイメージとしてディスク 70 に格納する方法ではデータ処理装置が作成された後にすべてのデータ処理装置に共通なプログラムイメージを追加することができるからである。また、ディスク上に格納することにより機能の追加や修正も容易に行うことができるようになる。

【 0 0 3 1 】

以上により、同一システム上でオープンシステムのオペレーティングシステムと別のアーキテクチャで動作する汎用コンピュータのオペレーティングシステムをオープンシステムのオペレーティングシステムに変更を加えることなくそれぞれ独立して同時に動作させることが可能になる。

【 0 0 3 2 】

（本発明の第二の実施の形態）

次に、本発明の第二の実施の形態について説明する。この実施の形態は第一の

実施の形態を拡張するものであり、プロセッサやメモリなどのハードウェア資源を増設した際に、それらのハードウェア資源を第一のオペレーティングシステムで使用するのか第二のオペレーティングシステムで使用するのかを容易に指示できるものである。

【 0 0 3 3 】

図 1 1 は、本発明の第二の実施の形態のハードウェア分割処理の手順を示すフローチャートである。

【 0 0 3 4 】

第一の実施の形態のハードウェア分割手段 2 が分割ハードウェア資源数量記憶手段 1 から分割するハードウェア資源についての情報を取得する前に、オペレータからの指示があるかどうかを確認する（ステップ C 1）。指示がある場合にはその指示にしたがってハードウェア分割手段 2 がハードウェア資源を分割するとともに（ステップ C 2）、分割ハードウェア資源数量記憶手段 1 によって分割したハードウェア資源の数量を記憶する（ステップ C 3）。指示がない場合には分割ハードウェア資源数量記憶手段 1 から分割するハードウェアの情報を取得する（ステップ C 4）。それぞれ得られた情報に基づいてハードウェア資源を分割する（ステップ C 5）。

【 0 0 3 5 】

以上の手順によりプロセッサやメモリなどのハードウェア資源を増設した際に増設したハードウェア資源を第一のオペレーティングシステムで使用するのか第二のオペレーティングシステムで使用するのかを容易に指示できることになる。また、負荷に応じて第一のオペレーティングシステムと第二のオペレーティングシステムで使用するハードウェア資源を容易に調整することができるようになる。

【 0 0 3 6 】

（本発明の第三の実施の形態）

次に、本発明の第三の実施の形態について説明する。この実施の形態は第一および第二の実施の形態を拡張するものであり、第一のオペレーティングシステムと第二のオペレーティングシステムが共通にアクセスできるメモリ領域を使用し

て両者のデータを交換できるようにするとともに、第一のオペレーティングシステムと第二のオペレーティングシステムとの間の通信機能を利用して、第二のオペレーティングシステム側から第一のオペレーティングシステムが管理する外部入出力装置を利用できるようにするものである。

【0037】

まず、共有メモリの確保について説明する。

【0038】

図12に示すように分割ハードウェア資源管理テーブル110に共有メモリ33に関するエントリ115を設ける。ハードウェア資源分割手段2は分割ハードウェア資源数量記憶手段1によって得られるこの情報に基づいてハードウェア資源を分割する。このとき図13に示すように第二オペレーティングシステム用ハードウェア資源管理テーブル310にメモリ管理テーブル313に共有メモリとして使用するサイズと共有メモリであることを示す属性を持つエントリを追加する。また、第一のオペレーティングシステムからアクセス可能なディスク70上のファイルまたは不揮発性メモリ45上に図14に示す共有メモリ管理テーブル400を格納する。第二のオペレーティングシステムから共有メモリにアクセスする手段としては、第二のオペレーティングシステムの初期化時にメモリ管理テーブル313に基づいて共有メモリに対してアクセスするためのページテーブルを用意し、入出力処理部700がこの共有メモリ用のページを介して共有メモリ33にアクセスできるようにする。

【0039】

第一のオペレーティングシステム上には共有メモリ33にアクセスする手段として第一オペレーティングシステムデバイスドライバ600を用意する。この第一オペレーティングシステムデバイスドライバ600はドライバの初期化時に共有メモリ管理テーブル400から共有メモリ33の開始アドレスおよびレンジの情報を取得して、この共有メモリ33をメモリにマップされた入出力空間として扱うものとする。

【0040】

以上の手順により共有メモリ33を介することによって物理的なネットワーク

機器を使用することなく第一のオペレーティングシステムと第二のオペレーティングシステムが高速にデータ交換を行うことができるようになる。

【 0 0 4 1 】

次に、第一のオペレーティングシステムと第二のオペレーティングシステムとの間の通信機能について説明する。

【 0 0 4 2 】

第一のオペレーティングシステムと第二のオペレーティングシステムとの間の通信手段として、本発明の一例では図 1 5 に示すように 1 つの装置で 2 つの割り込み信号をプロセッサに通知することができる通信用装置 9 0 を使用する。この通信用装置 9 0 は論理的に論理装置 9 1 と論理装置 9 2 の 2 つの装置を持っており、論理装置 9 1 に出力要求があると論理装置 9 2 がプロセッサに割り込みを通知し、論理装置 9 2 に出力要求があると論理装置 9 1 がプロセッサに割り込みを通知するものとする。ファームウェアはシステム起動時に図 1 6 に示すようにハードウェア資源管理テーブル内の割り込み管理テーブル 2 4 0 にこの外部入出力装置である通信用装置 9 0 のエントリ 2 4 5 と 2 4 6 を作成する。分割ハードウェア資源管理テーブル 1 1 0 では外部入出力装置である通信用装置 9 0 の論理装置 9 2 の割り込みを第二のオペレーティングシステムで使用するように第二オペレーティングシステム用割り込みエントリ 1 1 3 が格納されているものとする。ハードウェア資源分割手段 2 は分割ハードウェア資源数量記憶手段 1 によって得られるこの情報に基づいてハードウェア資源を分割する。

【 0 0 4 3 】

第二のオペレーティングシステムは、論理装置 9 2 への出力と同装置からの割り込みを処理する入出力処理部を持つものとする。

【 0 0 4 4 】

図 1 7 に示すように第一のオペレーティングシステム上には第一のオペレーティングシステム用の外部入出力装置である通信用装置 9 0 の論理装置 9 1 への出力と同装置からの割り込みを処理する第一オペレーティングシステムデバイスドライバ 6 0 0 を用意する。

【 0 0 4 5 】

第二のオペレーティングシステムから第一のオペレーティングシステムが管理する外部入出力装置である通信用装置 9 0 にアクセスする方法を説明する。

【 0 0 4 6 】

第二のオペレーティングシステムの入出力処理部 7 0 0 と第一のオペレーティングシステムの第一オペレーティングシステムデバイスドライバ 6 0 0 間では図 1 8 に示すようなコマンドテーブル 5 0 0 を使用する。コマンドテーブル 5 0 0 には入力か出力かを示すコマンドデータ 5 0 1 と第一のオペレーティングシステムで管理するデバイス番号 5 0 2 と入出力データを格納する共有メモリ 3 3 上のアドレス 5 0 3 と入出力データのレンジ 5 0 4 を格納するものとする。デバイス番号 5 0 2 は第二のオペレーティングシステムと第一のオペレーティングシステムのデバイスドライバであらかじめ決めてある番号を使用してもよいし、デバイス番号を管理するテーブルを用意しても良い。

【 0 0 4 7 】

以下、図 1 7 に示される動作の一例について説明する。第二のオペレーティングシステムから第一のオペレーティングシステムへ入力要求を発行するときには、コマンドテーブル 5 0 0 に入力を示すコマンドデータ、入力を要求するデバイス番号、入力データのアドレスおよびレンジを格納し（ステップ①）、論理装置 9 2 へ出力要求を出す（ステップ②）。論理装置 9 2 は論理装置 9 1 から第一のオペレーティングシステムの管理するプロセッサへ割り込みを通知する。第一オペレーティングシステムデバイスドライバ 6 0 0 がこの割り込みを受け付け（ステップ③）、コマンドを解析して（ステップ④）、要求のあったデバイスから入力処理を行う。入力データを指示されたアドレスに格納し（ステップ⑤）、論理装置 9 1 へ出力要求を行う（ステップ⑥）。論理装置 9 1 は論理装置 9 2 から第二のオペレーティングシステムの管理するプロセッサへ割り込みを通知する。第二のオペレーティングシステムの入出力処理部がこの割り込みを受け付け（ステップ⑦）、入力データを処理する（ステップ⑧）。

【 0 0 4 8 】

【発明の効果】

以上説明したように、本発明には以下の効果がある。

【 0 0 4 9 】

本発明によれば第二のオペレーティングシステムとして汎用コンピュータのエミュレーションプログラムを使用すれば特殊なハードウェアを必要とすることなく、安価にオープンシステムと汎用コンピュータシステムを同一計算機上で共存させることができるという効果がある。また、オープンシステムのオペレーティングシステムとは独立して汎用コンピュータのオペレーティングシステムをオープンシステムのプロセッサ上で直接実行できるため汎用コンピュータのエミュレーションシステムを高速に実行できる。また、メモリを介して両者のデータを交換できるためシステム間の通信を高速に行うことができる。

【図面の簡単な説明】

【図 1】

本発明の第一の実施の形態のシステム構成を示すブロック図である。

【図 2】

本発明の第一の実施の形態の構成を示すブロック図である。

【図 3】

本発明の第一の実施の形態の分割ハードウェア資源管理テーブルの構成を示す図である。

【図 4】

本発明の第一の実施の形態のハードウェア資源管理テーブルの構成を示す図である。

【図 5】

本発明の第一の実施の形態のハードウェア資源管理テーブルのうちプロセッサ管理テーブルの構成を示す図である。

【図 6】

本発明の第一の実施の形態のハードウェア資源管理テーブルのうち割り込み管理テーブルの構成を示す図である。

【図 7】

本発明の第一の実施の形態のハードウェア資源管理テーブルのうちメモリ管理テーブルの構成を示す図である。

【図 8】

本発明の第一の実施の形態のエミュレーションハードウェア資源管理テーブルの構成を示す図である。

【図 9】

従来技術におけるシステム起動時の動作を示すフローチャートである。

【図 1 0】

本発明の第一の実施の形態におけるシステム起動時の動作を示すフローチャートである。

【図 1 1】

本発明の第二の実施の形態のハードウェア分割処理の手順を示すフローチャートである。

【図 1 2】

本発明の第三の実施の形態の分割ハードウェア資源管理テーブルの構成を示す図である。

【図 1 3】

本発明の第三の実施の形態のエミュレーションハードウェア資源管理テーブルの構成を示す図である。

【図 1 4】

本発明の第三の実施の形態の共有メモリ管理テーブルの構成を示す図である。

【図 1 5】

本発明の第三の実施の形態の通信用装置の構成を示す図である。

【図 1 6】

本発明の第三の実施の形態のハードウェア資源管理テーブルのうち割り込み管理テーブル構成を示す図である。

【図 1 7】

本発明の第三の実施の形態の構成を示すブロック図である。

【図 1 8】

本発明の第三の実施の形態のコマンドテーブルの構成を示す図である。

【符号の説明】

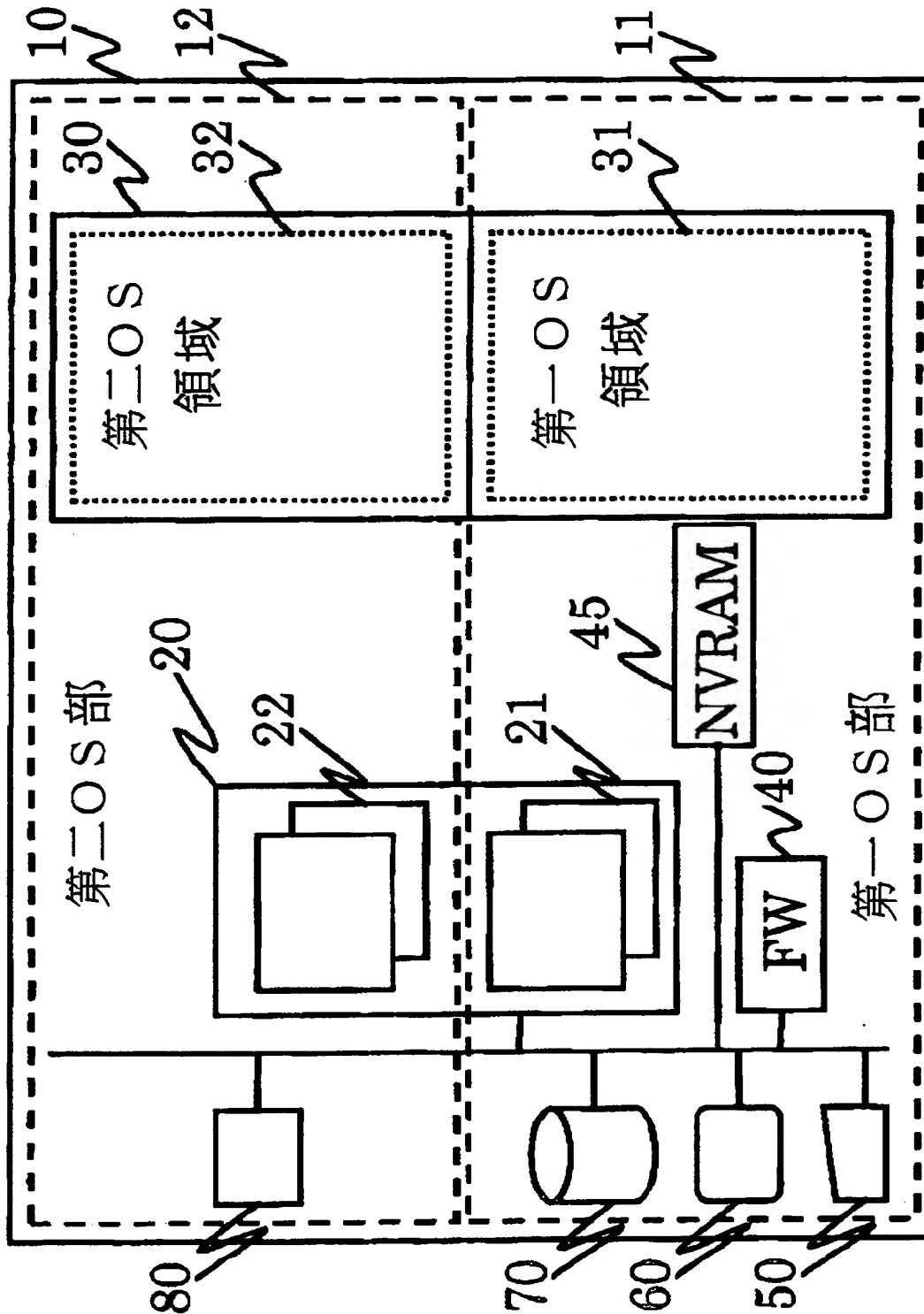
- 1 分割ハードウェア資源数量記憶手段
- 2 ハードウェア資源分割手段
- 3 第二オペレーティングシステム起動手段
- 4 第一オペレーティングシステム起動手段
- 1 0 コンピュータ
- 1 1 第一オペレーティングシステム部
- 1 2 第二オペレーティングシステム部
- 2 0 プロセッサ
- 2 1 第一オペレーティングシステム用プロセッサ
- 2 2 第二オペレーティングシステム用プロセッサ
- 3 0 メモリ
- 3 1 第一オペレーティングシステム用メモリ
- 3 2 第二オペレーティングシステム用メモリ
- 3 3 共有メモリ
- 4 0 記憶装置
- 4 5 不揮発性メモリ
- 5 0 キーボード
- 6 0 ディスプレイ
- 7 0 ディスク
- 8 0 外部入出力装置
- 9 0 通信用装置
- 9 1 論理装置 1
- 9 2 論理装置 2
- 1 0 0 分割ハードウェア資源管理テーブル
- 1 0 1 第二オペレーティングシステム用プロセッサエントリ
- 1 0 2 第二オペレーティングシステム用割り込みエントリ
- 1 0 3 第二オペレーティングシステム用メモリエントリ
- 1 1 0 分割ハードウェア資源管理テーブル
- 1 1 1 第二オペレーティングシステム用プロセッサエントリ

1 1 2	第二オペレーティングシステム用割り込みエントリ 1
1 1 3	第二オペレーティングシステム用割り込みエントリ 2
1 1 4	第二オペレーティングシステム用メモリエントリ
1 1 5	共有メモリエントリ
2 0 0	ハードウェア資源管理テーブル
2 1 0	プロセッサ管理テーブル
2 1 1	プロセッサエントリ 1
2 1 2	プロセッサエントリ 2
2 1 3	プロセッサエントリ 3
2 1 4	プロセッサエントリ 4
2 2 0	割り込み管理テーブル
2 2 1	割り込みエントリ 1
2 2 2	割り込みエントリ 2
2 2 3	割り込みエントリ 3
2 2 4	割り込みエントリ 4
2 3 0	メモリ管理テーブル
2 3 1	メモリエントリ 1
2 3 2	メモリエントリ 2
2 3 3	メモリエントリ 3
3 0 0	第二オペレーティングシステム用ハードウェア資源管理テーブル
3 0 1	プロセッサ管理テーブル
3 0 2	割り込み管理テーブル
3 0 3	メモリ管理テーブル
3 1 0	第二オペレーティングシステム用ハードウェア資源管理テーブル
3 1 1	プロセッサ管理テーブル
3 1 2	割り込み管理テーブル
3 1 3	メモリ管理テーブル
4 0 0	共有メモリ管理テーブル
5 0 0	コマンドテーブル

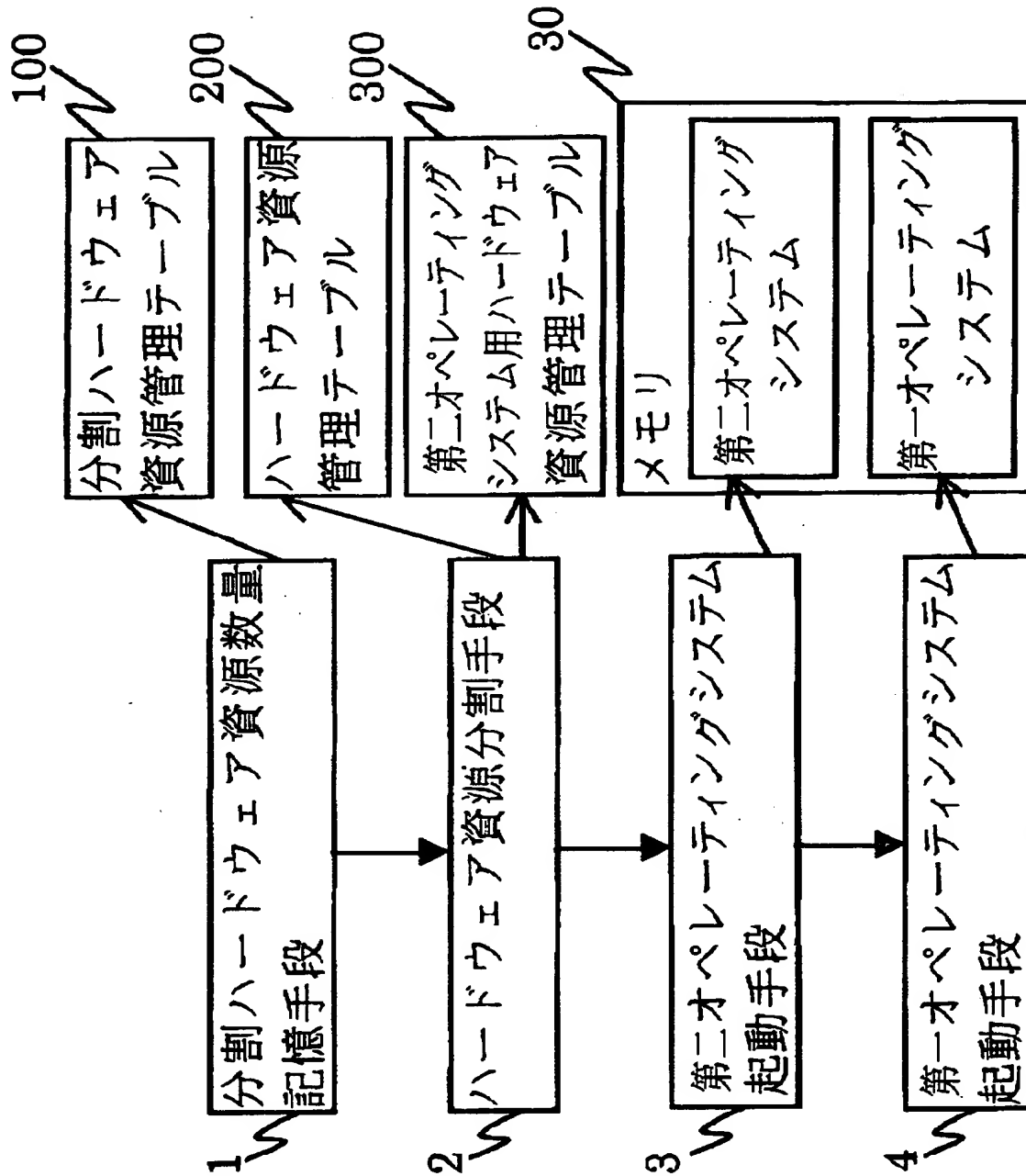
5 0 1	コマンドデータ
5 0 2	デバイス番号
5 0 3	データアドレス
5 0 4	データレンジス
5 1 0	データ領域
6 0 0	第一オペレーティングシステムデバイスドライバ
7 0 0	第二オペレーティングシステム入出力処理部

【書類名】 図面

【図 1】



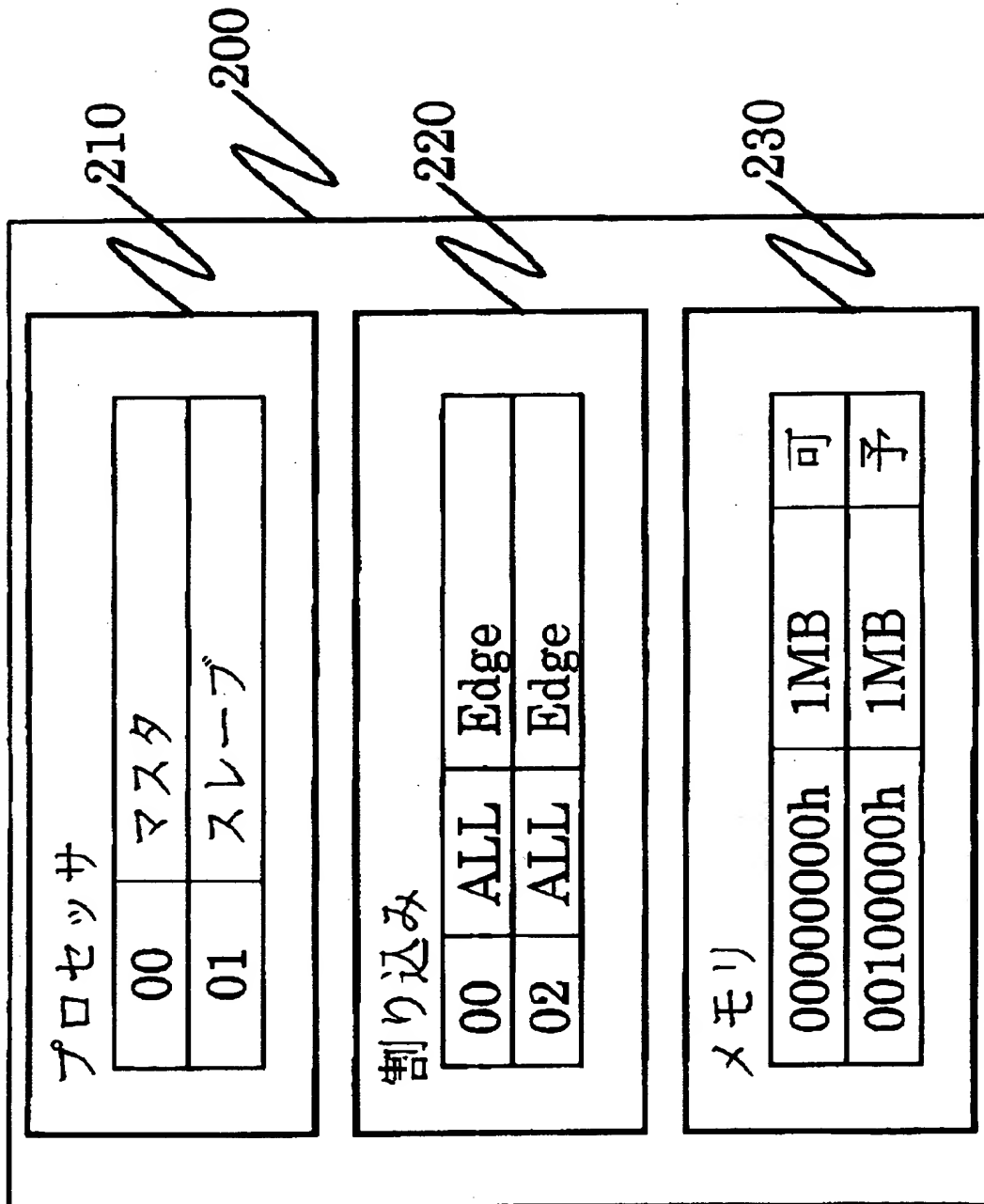
【図2】



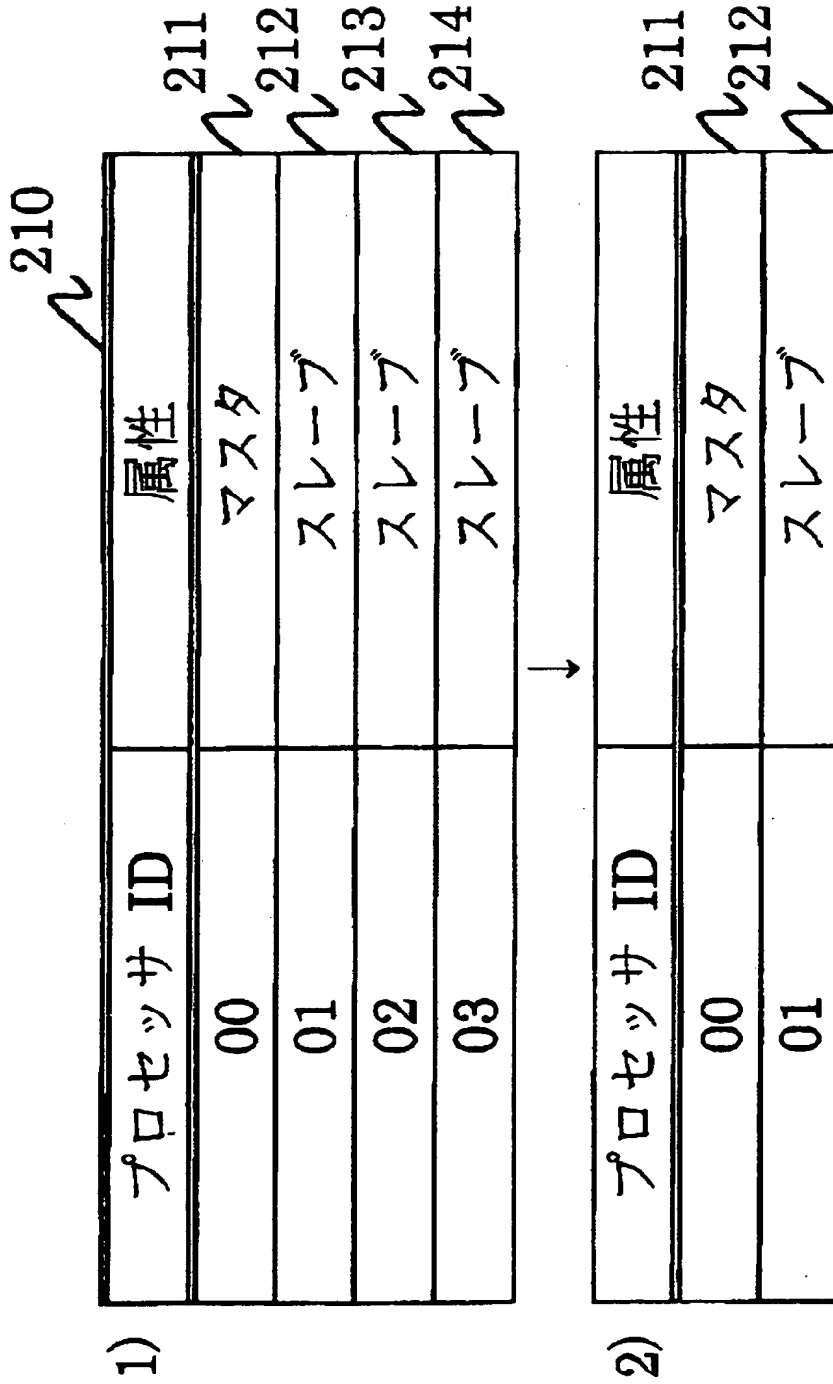
【図 3】

ハードウェア資源	数量
プロセッサ	2
割り込み	80
メモリ	256MB

【図4】



【図 5】



【図 6】

220

1)

デバイス ID	割込番号	通知先	属性
50	01	すべて	エッジ
60	05	すべて	エッジ
80	10	すべて	レベル
70	06	すべて	エッジ

221 222 223 224

↓

2)

デバイス ID	割込番号	通知先	属性
50	01	すべて	エッジ
60	05	すべて	エッジ
70	06	すべて	エッジ

221 222 224

【図 7】

230

1)

開始アドレス	レンジス	属性
00000000h	1MB	使用可能
00100000h	1MB	予約済
00200000h	510MB	使用可能

231 232 233

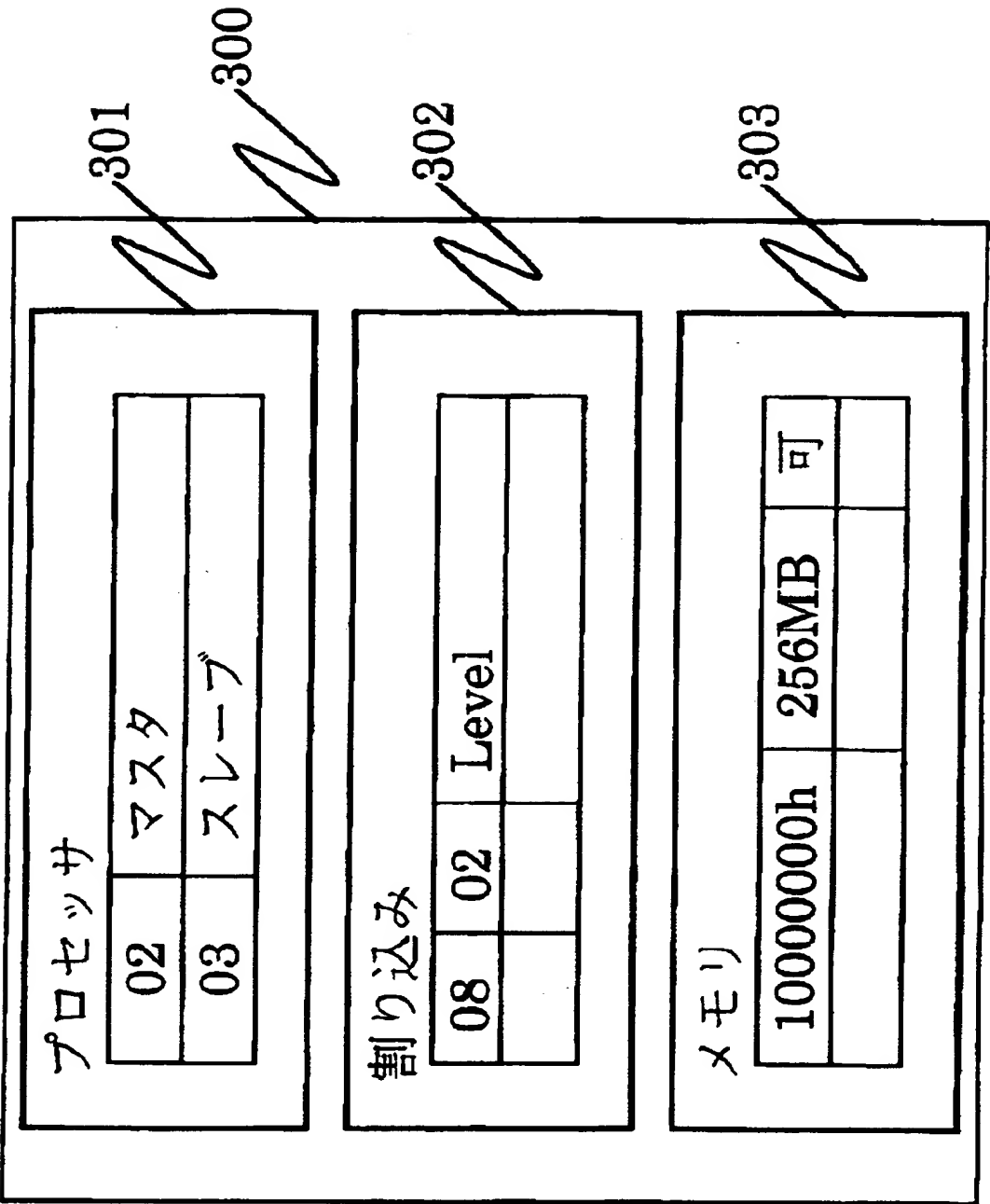
↓

2)

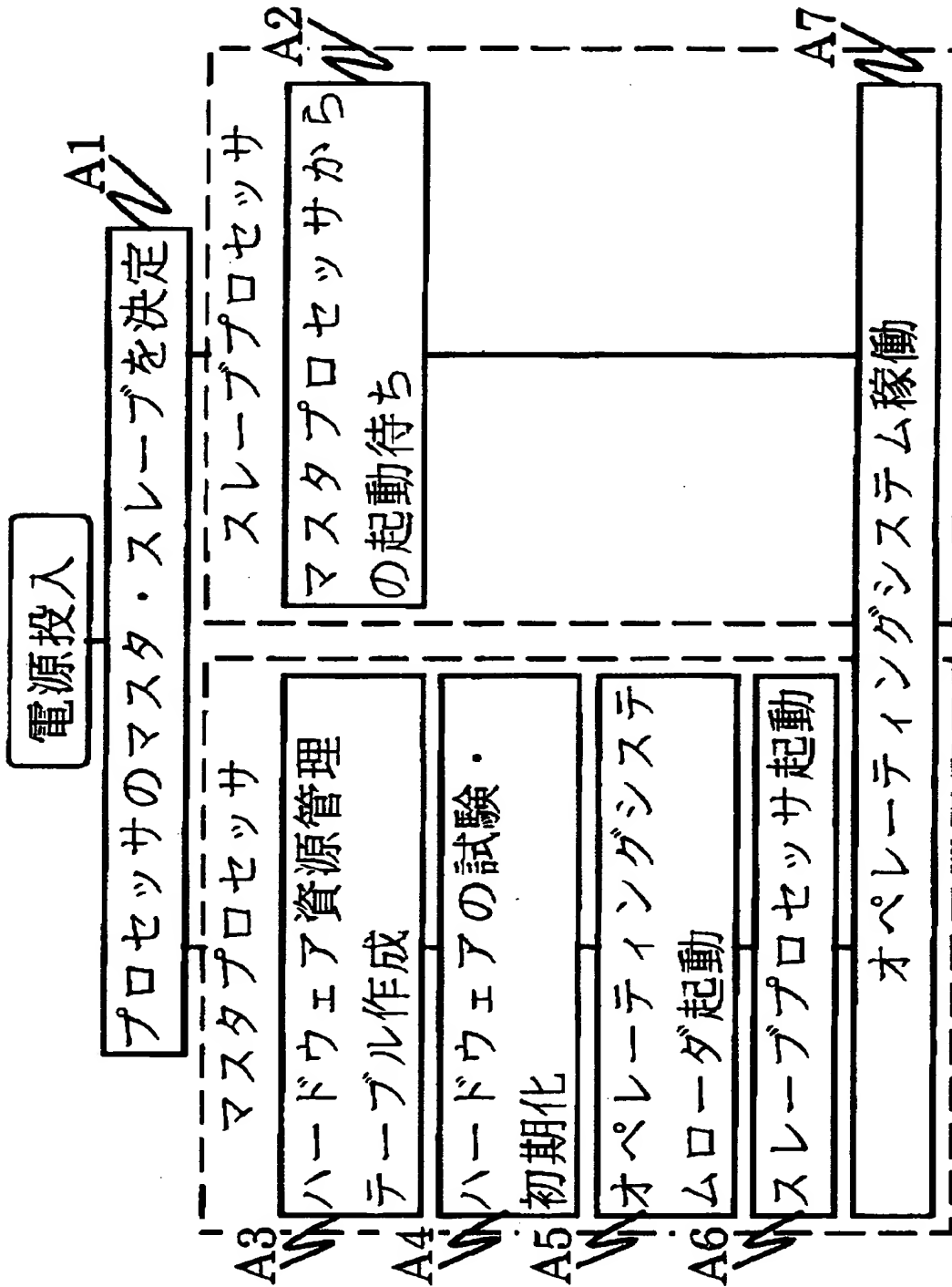
開始アドレス	レンジス	属性
00000000h	1MB	使用可能
00100000h	1MB	予約済
00200000h	254MB	使用可能

231 232 233

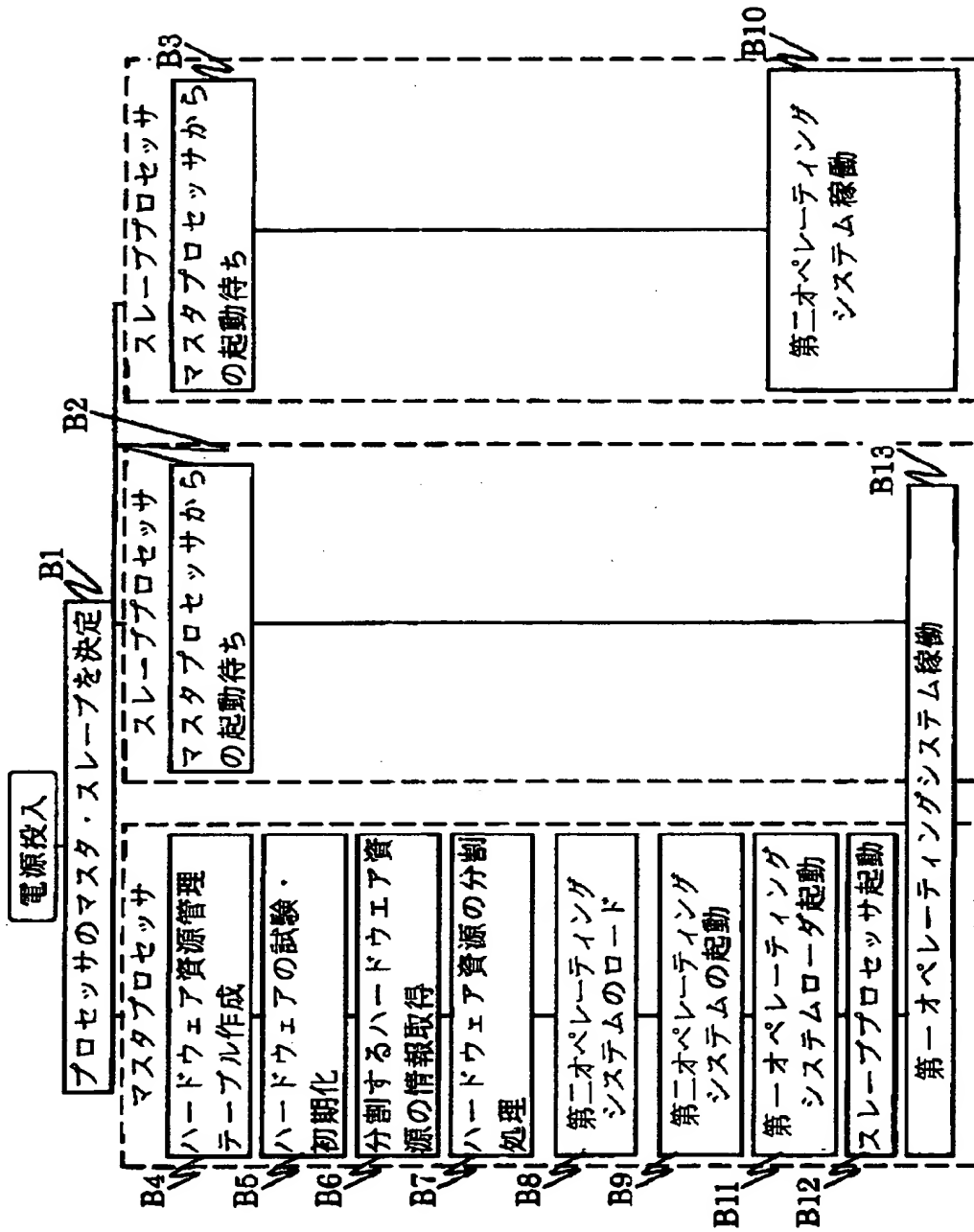
【図 8】



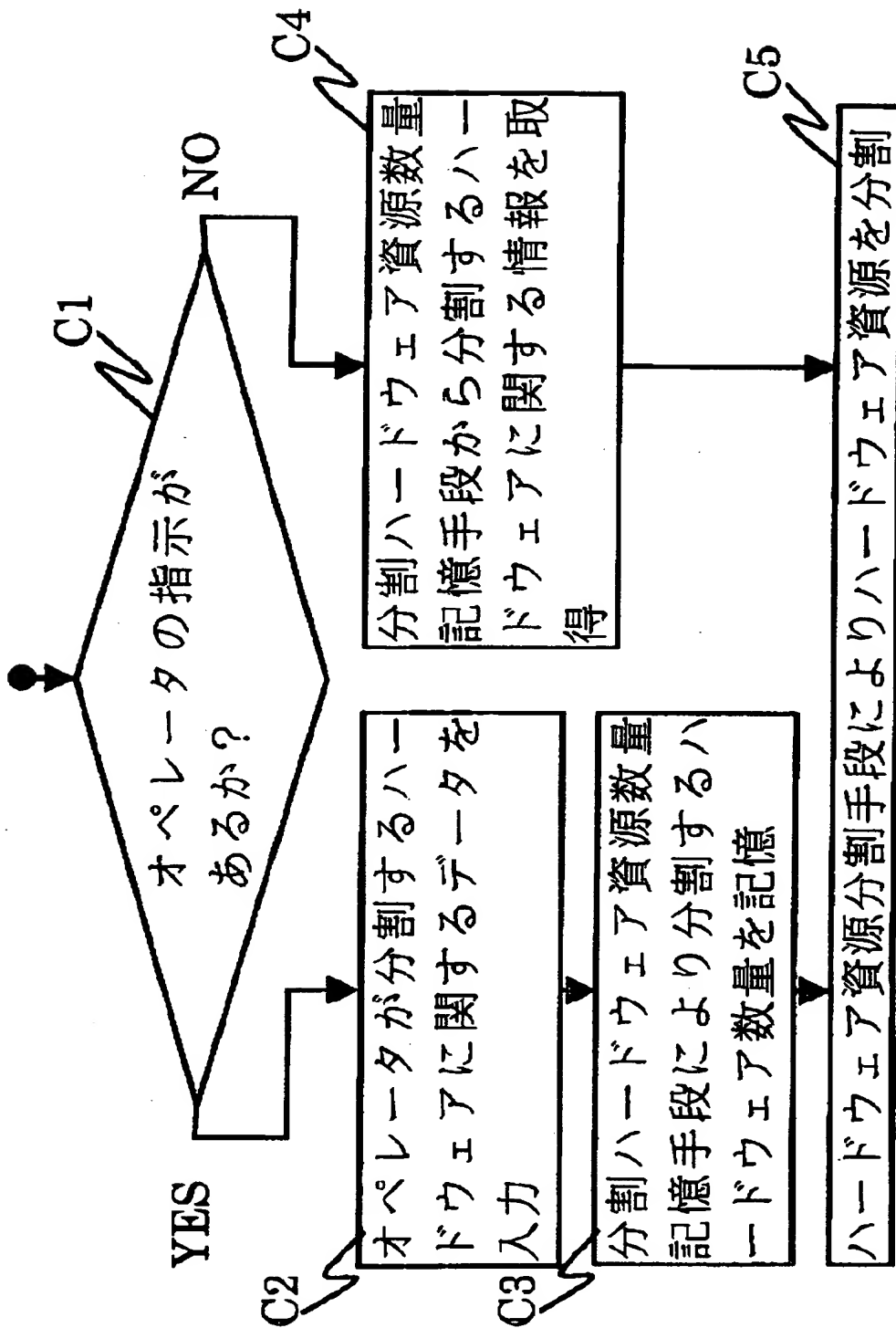
【図9】



【図10】



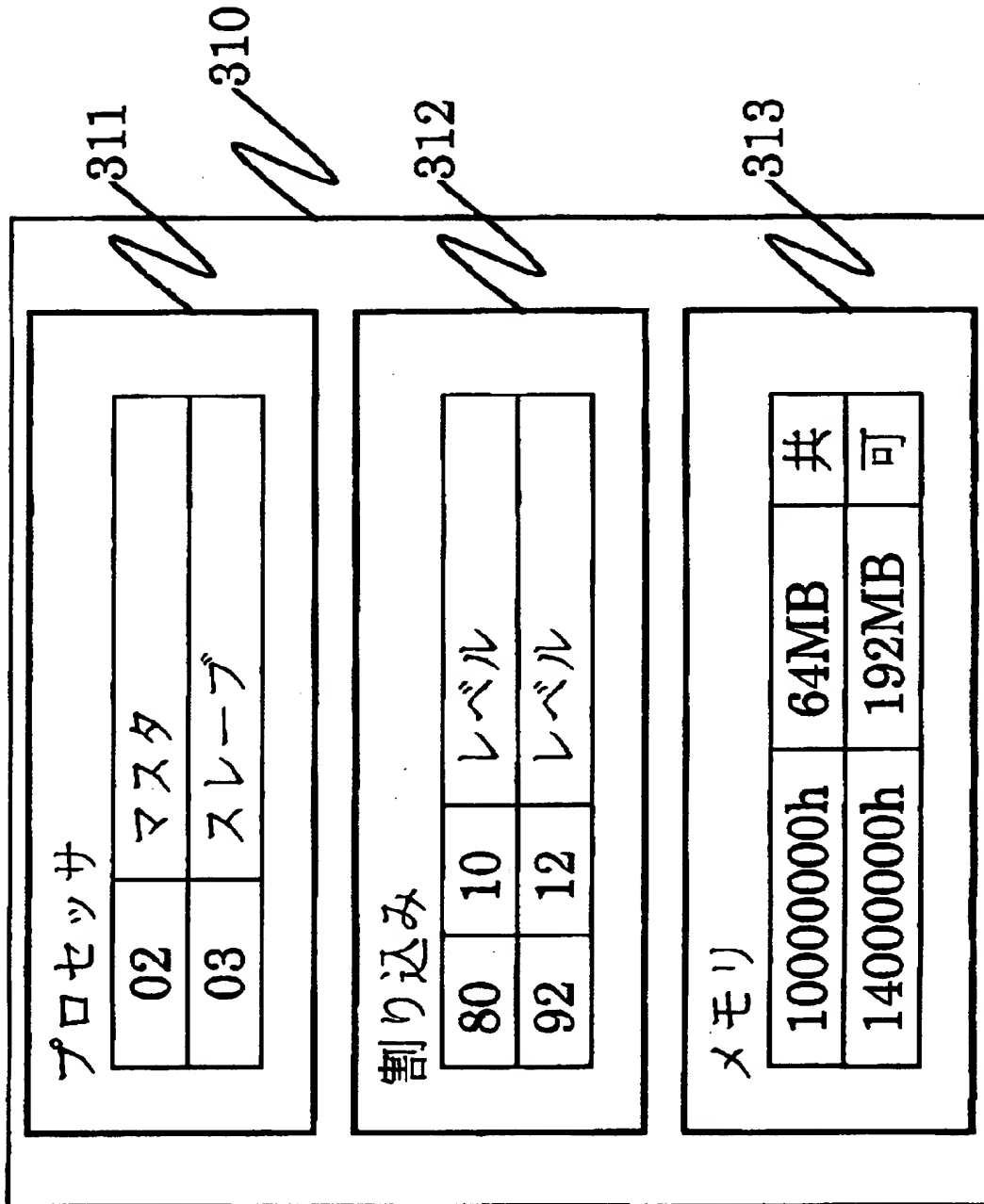
【図 11】



【図 12】

ハードウェア資源	数量
プロセッサ	2
割り込み	80
割り込み	92
メモリ	192MB
共有メモリ	64MB

【図13】

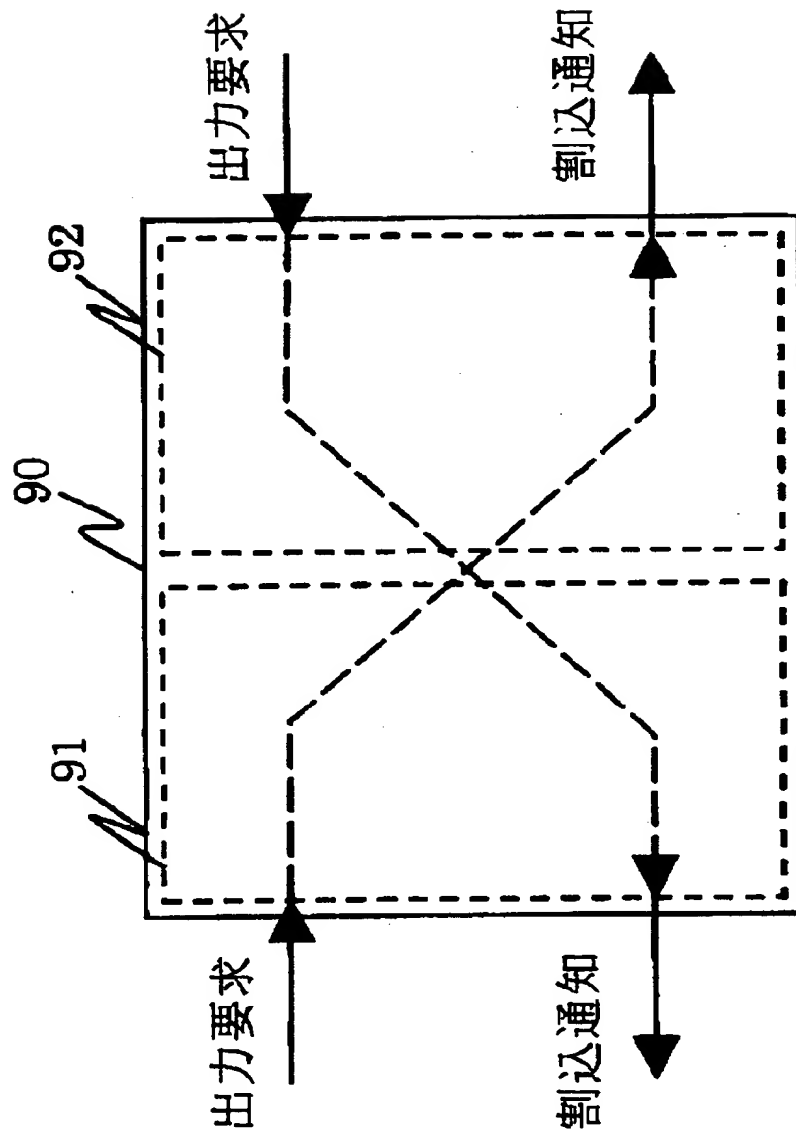


【図 1 4】

400 ↗

開始アドレス	レンジス	属性
10000000h	64MB	共通

【図 1 5】



【図16】

1) 240

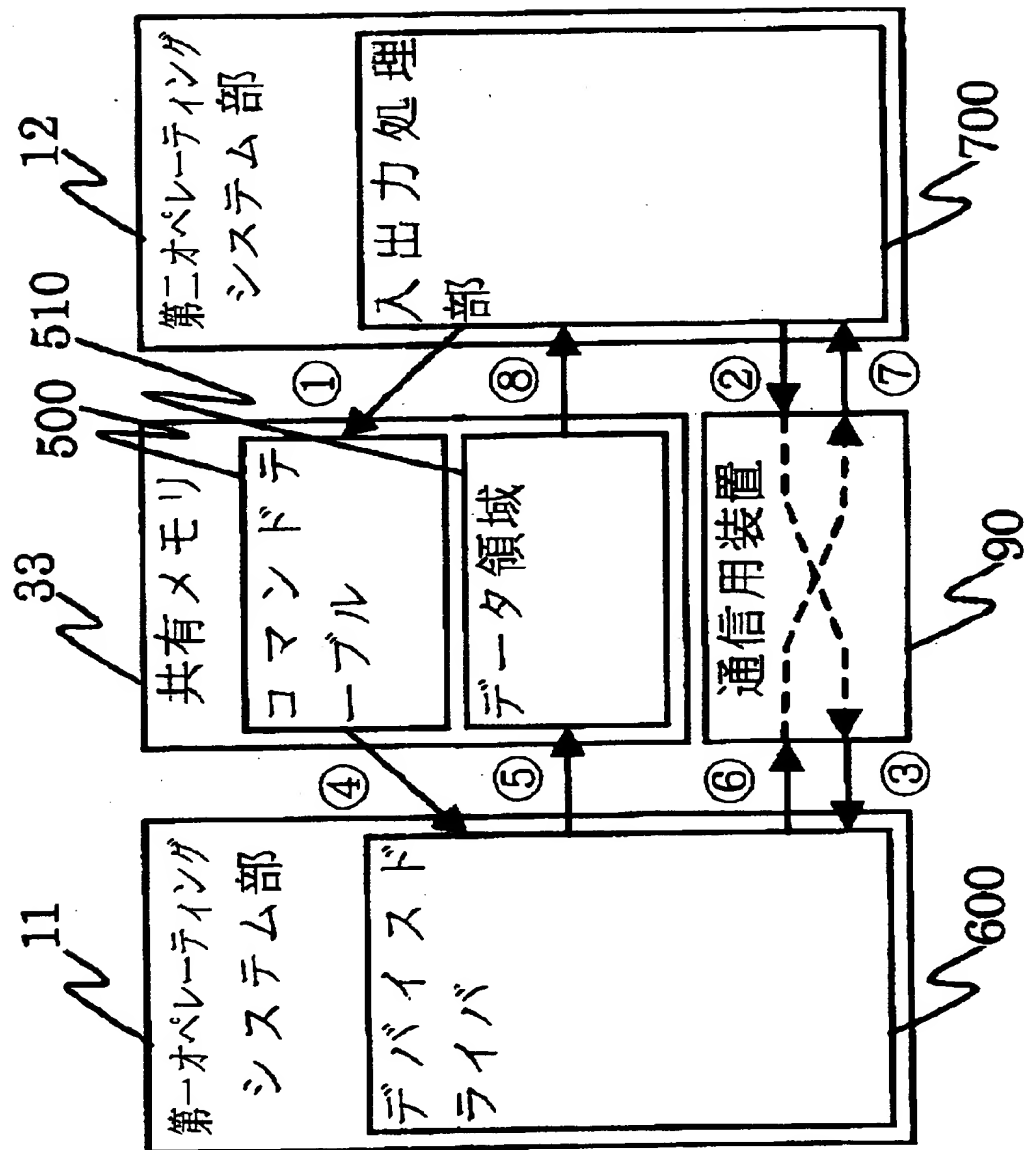
デバイスID	割込番号	通知先	属性
50	01	すべて	エッジ
60	05	すべて	エッジ
80	10	すべて	レベル
70	06	すべて	エッジ
91	11	すべて	レベル
92	12	すべて	レベル

↓

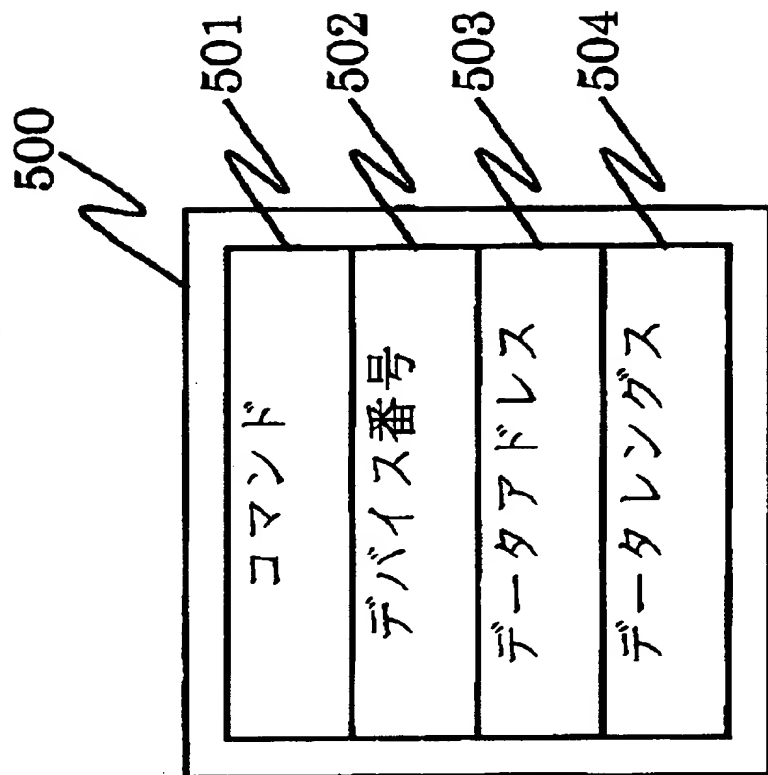
2)

デバイスID	割込番号	通知先	属性
50	01	すべて	エッジ
60	05	すべて	エッジ
70	06	すべて	エッジ
91	11	すべて	レベル

【図17】



【図18】



【書類名】 要約書

【要約】

【課題】 安価で高速なオープンシステムと汎用コンピュータシステムとを融合したシステムを実現するヘテロジーニアスなプロセッサシステムを提供する。

【解決手段】 第一のオペレーティングシステムの起動前にハードウェア資源分割手段 2 によりハードウェア資源管理テーブル 2 0 0 から第二のオペレーティングシステムの動作に必要なハードウェア資源を分割し、第二オペレーティングシステム起動手段 3 により第二のオペレーティングシステムを動作させた後に第一オペレーティングシステム起動手段 4 により第一のオペレーティングシステムを稼働させる。

【選択図】 図 2

出 願 人 履 歴 情 報

識別番号 [000242666]

1. 変更年月日	1990年 8月28日
[変更理由]	新規登録
住 所	石川県石川郡鶴来町安養寺1番地
氏 名	北陸日本電気ソフトウェア株式会社